Using an interior point cutting plane method to solve integer programming problems — Final report of research supported by ONR Grant number N00014–90–J–1714.

John E. Mitchell
Department of Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, NY 12180

September 30, 1992

Abstract

There were several accomplishments of this research, both theoretical and computational. In joint work with Todd, we presented a cutting plane primal projective interior point method which we applied to matching problems, with encouraging computational results. Primal projective methods require a method to update the dual; we showed how various dual updates are related to each other and we also derived a dual projective algorithm. We derived a polynomial-time shifted barrier warm start algorithm which can be used in a cutting plane method; we showed that the directions obtained are strongly related to the directions derived in the work with Todd; computational results showed that the algorithm can be useful in some situations. The grant partially supported a Ph.D. student, Brian Borchers, who received his degree in August, 1992. His thesis concerned the use of branch-and-bound methods and contained good computational results as well as interesting theoretical observations. One paper from this thesis describes how the primal-dual interior point method can be used efficiently in a branch-and-bound method for solving mixed integer linear programming problem. Another paper describes how branch and bound algorithms for nonlinear integer programming problems can be improved. Borchers and I also developed a primal-dual interior point cutting plane method for solving linear ordering problems; the computational results for this algorithm were very encouraging, with run times comparable to those required by a simplex based cutting plane algorithm.

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY N/A	3. DISTRIBUTION / AVAILABILITY OF REPORT Unclassified
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION RPI 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Department of Mathematical Sciences Rensselaer Polytechnic Institute Troy, NY 12180-3590	7b. ADDRESS (City, State, and ZIP Code) Office of Naval Research 800 N. Quincy St. Arlington, VA 22217-5000
8a. NAME OF FUNDING / SPONSORING ORGANIZATION ONR 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N 00014-90-J-1714
8c. ADDRESS (City, State, and ZIP Code) Same as 7b	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
Using an interior point cutting plane method to solve integer programming problems

12. PERSONAL AUTHOR(S)
John E. Mitchell

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM 6/1/92 TO 9/30/92	September 30, 1992	12

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

(See Attached)

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ OTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (Include Area Code) / 22c. OFFICE SYMBOL

DD Form 1473, JUN 86 *Previous editions are obsolete.* SECURITY CLASSIFICATION OF THIS PAGE

19.

Abstract

There were several accomplishments of this research, both theoretical and computational. In joint work with Todd, we presented a cutting plane primal projective interior point method which we applied to matching problems, with encouraging computational results. Primal projective methods require a method to update the dual; we showed how various dual updates are related to each other and we also derived a dual projective algorithm. We derived a polynomial-time shifted barrier warm start algorithm which can be used in a cutting plane method; we showed that the directions obtained are strongly related to the directions derived in the work with Todd; computational results showed that the algorithm can be useful in some situations. The grant partially supported a Ph.D. student, Brian Borchers, who received his degree in August, 1992. His thesis concerned the use of branch-and-bound methods and contained good computational results as well as interesting theoretical observations. One paper from this thesis describes how the primal-dual interior point method can be used efficiently in a branch-and-bound method for solving mixed integer linear programming problem. Another paper describes how branch and bound algorithms for nonlinear integer programming problems can be improved. Borchers and I also developed a primal-dual interior point cutting plane method for solving linear ordering problems; the computational results for this algorithm were very encouraging, with run times comparable to those required by a simplex based cutting plane algorithm.

1 Introduction

This is the final report of the research supported by the ONR Grant number N00014–90–J–1714. In this report we summarize the research accomplished and the papers produced.

Currently, almost all approaches to solving integer programming problems with linear programming methodology use the simplex method to solve the linear programs. There have been several notable successes with such algorithms. Within the last eight years, interior point methods have become accepted as powerful tools for solving linear programming problems. It appears that interior point methods may well solve large linear programs substantially faster than the simplex method. A natural question, therefore, is whether interior point methods can be successfully used to solve integer programming problems. This was the subject of the research conducted under this grant.

We are interested in branch-and-cut algorithms, which are a combination of cutting plane methods and branch-and-bound methods. Let S be the convex hull of the set of feasible integer points. In a traditional cutting plane method, the linear programming relaxation of the problem is solved to optimality using the simplex algorithm: if the optimal solution $\bar{x}$ is integer, then we are done; otherwise $\bar{x}$ can be *separated* from S, an extra constraint (or *cutting plane*) added to the relaxation and the process repeated. The extra constraint takes the form $a_k^T x \leq b_k$; this constraint is satisfied by all points in S but it is violated by $\bar{x}$. Traditional cutting plane methods used Gomory cuts, but the recent success of cutting plane methods has come about with the use of facet-defining inequalities, which give proper faces of maximal dimension of the convex hull of the set of feasible integer points (see, for example, [6, 10, 11, 12, 13, 23]). We use an interior point method in place of the simplex algorithm. We usually do not solve the relaxation to optimality, but attempt to find cutting planes before reaching optimality. This is usually possible with an interior point method; in fact, it is more attractive with an interior point method than the simplex method because the interior point method gets close to optimality quickly, whereas the simplex method may pivot a vital element into the basis on the last iteration.

The difficulty with using an interior point method in a cutting plane algorithm is that the solution to one relaxation is usually not a good starting point for an interior point method, because it is close to the boundary of the feasible region. Thus, it is usually necessary to attempt to stop solution early: the earlier we are able to find good cutting planes, the better the initial solution to the next relaxation. Early termination obviously reduces the number of iterations spent solving the current relaxation, but in addition it reduces the number of iterations spent solving the next relaxation, because the initial point to the next relaxation is more centered. There

are two potential disadvantages to trying to find cutting planes early: if the search for cutting planes is unsuccessful, we have wasted time; secondly, it may well be that superfluous constraints are added, with the result that the algorithm requires extra iterations and extra stages of adding cutting planes. The attempt to exploit early termination is a recurrent theme of the research generated under this grant.

The grant partially supported the paper with Todd [22] which described a primal projective cutting plane algorithm, which was used to solve matching problems. The primal projective method was applied to the dual of the current relaxation of the integer programming problem. We described how cutting planes can be added and also how variables can be added. We gave computational results which illustrated that such an algorithm may be attractive, provided cutting planes are identified early. We describe this paper in more detail in Section 2.

Primal projective algorithms require a method for updating the dual. In [18], we showed how two dual updates are related to each other. We also described a dual projective algorithm, deriving the direction by using a QR-decomposition. This paper is the subject of Section 3.

When adding a cutting plane, we add a variable to the dual. If we give this variable value zero, then the new dual solution is feasible. Unfortunately, this is not a good starting point for an interior point method because interior point methods generally require that all variables should be strictly positive. In [17], we described how shifted barriers can be used to overcome this difficulty. The barriers are the nonnegativity constraints; if these are relaxed slightly, or shifted, then giving the new variable the value zero gives a point which strictly satisfies the inequality constraints; this is now a feasible starting point for an interior point method. This algorithm is discussed further in Section 4.

This grant partially supported the research of the student Brian Borchers, who received his Ph.D. in August 1992 [3]. One aspect of his thesis was to describe a primal-dual barrier method for solving mixed integer linear programming problems using branch-and-bound [5]. This research showed that an interior point method can be employed successfully in a branch-and-bound algorithm because it is usually sufficient to obtain only an approximate solution at each node of the branch-and-bound tree. If it is necessary to branch on the current node, then the approximate solution provides a good initial solution for use in an interior point method to solve the child subproblem. Encouraging computational results were given. A spin off of our investigations into using interior point methods in this way was the observation that branch-and-bound methods for mixed integer nonlinear programming problems can also be improved through the use of early branching [4]. The work in Borchers's thesis is described in more detail in Section 5.

Borchers and I also developed a primal-dual barrier method cutting plane algorithm for solving linear ordering problems [19, 20]. The computational results we

obtained with this algorithm were comparable with those obtained using a simplex-based cutting plane method. These computational results give hope that interior point cutting plane methods may well outperform simplex methods on large problems, that is, problems which are not solvable on current hardware in a reasonable amount of time. With the rate at which hardware is improving, these currently intractable problems may soon be solvable with a cutting plane method based on an interior point method. For more details on this algorithm, see Section 6.

This grant has also partially supported the research of another student, Zhao-Yang Cheng, who is expected to receive his Ph.D. in the summer of 1993. Cheng has investigated the links between various interior point methods. He has also shown how interior point methods can be used to solve geometric programming problems; one of the methods used for these problems is a column generation approach.

2 Initial work using the primal projective method

In [22], we presented a cutting plane method for solving integer programming problems. This algorithm applied the primal projective method to the dual of the current relaxation of the integer programming problem. The dual solution was updated by using a method based upon the Todd-Burrell update [25].

There are several advantages to making the relaxation the dual problem when using the primal projective method. Primal feasibility is maintained even when cutting planes are added, so the method can be immediately restarted. The initial primal point, immediately after adding cutting planes to the dual, has all the additional primal variables set to zero. For the interior point algorithm to proceed, the variables must be strictly positive. The paper [22] gave a direction which is guaranteed to result in a strictly positive feasible point, if one exists. It was shown that moving in this direction was equivalent to solving a Phase I problem involving an artificial variable, in that the point generated by the solution to the Phase I problem corresponded to taking a certain step length in the direction. The primal projective algorithm for linear programming only uses the value of the dual solution when calculating the primal direction, so it does not need a dual solution which is strictly feasible. When the dual problem is a relaxation of the integer programming problem, any feasible solution to the integer program gives a feasible dual point, but this point is not usually strictly feasible. Because of the choice of algorithm, these solutions are immediately useful.

When setting up the initial relaxation of an integer programming problem, it is often useful to omit many of the variables. For example, for the perfect matching problem, the initial relaxation may contain variables only for the shortest ten edges adjacent to each vertex — see, for example, Grötschel and Holland [10]. After obtain-

ing an optimal solution to this revised problem using a cutting plane algorithm, it is necessary to check the reduced costs of the omitted variables. If some of the reduced costs are of the wrong sign, variables should be added to the formulation and the problem resolved. We gave a Phase I problem which can be used to initialize when these variables are added to the dual problem, and we gave a probabilistic argument to show that the Phase I problem is likely to be solved in one iteration when only a few variables are added.

The algorithm was implemented and used to solve perfect matching problems. We chose to experiment on the perfect matching problem because good separation heuristics for finding violated cutting planes are available [10]. The separation routines can only be called when the dual solution is successfully updated using the Todd-Burrell procedure. We found that it was best to call the separation routines as soon as the dual solution was updated; this dual solution was good enough that it was almost always possible to find violated cutting planes, provided the solution to the current relaxation did not solve the integer programming problem. The results were encouraging for two reasons. First, the number of iterations required at each relaxation was approximately six for the problems we tested, which is far fewer than the number required to solve the linear program from scratch, because cutting planes were identified before optimality and because the solution to the previous relaxation did serve as a warm start which could be exploited by the primal projective method. Second, the number of stages of adding cutting planes was limited, illustrating that the separation routines work well at an interior point.

3 Dual updates

The algorithm described in [22] updates the dual solution by solving a one-variable constrained version of the dual. In [24], Todd proposed an alternative update which involves solving a two-variable problem. In [9], Gonzaga motivated a one-variable update in a different manner. We showed that Todd's two-variable update and Gonzaga's one-variable update are equivalent in [18].

In [18], we also described a dual projective algorithm, deriving the directions through consideration of a QR-decomposition. Consideration of a QR-decomposition is a useful tool for analyzing projections when the constraint matrix can be broken into parts; some of the results in [21] were obtained through this technique.

4 Shifted barriers

In [17], we described a column-generation algorithm which is based upon Freund's two "warm start" shifted barrier function algorithms for linear programming [8]. By

using shifted barriers, extra variables can be introduced with value zero, and the new point will be an interior point.

In [8], the nonnegativity constraints $x \geq 0$ are replaced by constraints of the form $x + h(c^T x - B) \geq 0$, where $c^T x - B$ is the current duality gap and h is a nonnegative vector. Thus, if the vector h is chosen appropriately, a certain amount of negativity is allowed in the variables x. One result of this is that it is possible to have "interior" points where the variables are zero. This can be used in a column generation method: additional variables can be given the value zero when they enter.

Generating columns corresponds to adding constraints in the dual linear program. In a cutting plane algorithm for solving integer programming problems, constraints are added to a linear program as the solution proceeds. As was the case with the algorithm given in [22], this algorithm can be used when the relaxation of the integer programming problem is regarded as the dual linear programming problem. When cutting planes are added, the additional primal variables are given the value zero and the barrier constraints are satisfied strictly, so the interior point method can be restarted immediately, with no need for a Phase I procedure.

We analyzed this algorithm in the context of solving a linear program $\min\{c^T x : Ax = b, x \geq 0\}$, where we only work with a subset of the variables, adding extra variables as necessary, by checking for violation of the corresponding dual constraints. We showed that the resulting algorithm requires $O(nL)$ iterations, where n is the number of variables in the complete primal problem and L is the size of the data of the problem. We analyzed the direction taken by the algorithm immediately after a variable is added. We showed that this direction is a combination of three directions: the affine descent direction, the centering direction, and the direction obtained in [22]. Thus, this algorithm could be regarded as a refinement of the algorithm given in [22], with the algorithm generating a direction which is a combination of the three natural directions. For more information on the affine and centering directions, see, for example, the survey in den Hertog and Roos [14].

We implemented the algorithm and compared it with two other algorithms on some randomly generated transportation problems. The shifted barrier column generation algorithm performed considerably better than the primal projective algorithm applied to the full set of columns. We also compared the algorithm with a column generation algorithm similar to the one described in [22]: when only a few columns are added at once, it appears that the shifted barrier algorithm outperforms the other column generation algorithm, but the shifted barrier algorithm is slightly worse when many columns are added.

5 Branch-and-bound

The grant partially supported the research of Brian Borchers, who completed his Ph.D. in August 1992 [3]. His thesis was concerned with ways in which early termination could improve branch-and-bound algorithms.

In [5], we described a branch-and-bound algorithm which solved the linear programming subproblems by using the primal-dual barrier method. When using branch-and bound, one of four things can happen at the current node of the branch-and-bound tree. The subproblem could be unbounded; in an interior point method this can be detected by finding a ray in the dual problem. The subproblem could have optimal value worse than the value of a known integer feasible solution, so the node is fathomed by bounds; in an interior point method, this can usually be detected well before the subproblem is solved to optimality. The optimal solution could be an integer solution with value better than the best known solution; in this case we need to solve the subproblem to optimality, but the node is then fathomed. The final possibility is that the optimal solution to the subproblem has optimal value smaller than the best known solution, but the optimal solution is not feasible in the integer program; in this case, we can use heuristics based upon the basis identification techniques described in El-Bakry et al. [7] to determine that one of the integer variables is tending to a fractional value, and therefore that we should branch early.

It should be noted that in only one case is it necessary to actually solve the relaxation to optimality, and in that case the node is fathomed. When we branch early, one dual constraint is dropped, so the previous dual solution is still feasible. One primal variable is fixed, so infeasibilities are introduced into the primal problem. We found that we could still solve the child node quickly, provided we centered the solution slightly when restarting by adding small amounts to variables close to their bounds.

We solved several sets of problems using our branch-and-bound algorithm, comparing our results with the IBM package OSL [15]. Our algorithm was very competitive with OSL on some randomly generated capacitated facility location problems; these problems had no more than 20 warehouses and 400 destinations. Our algorithm did not perform so well on some test problems supplied to us by AT&T, on some problems drawn from the test set ORLIB [1], and on some problems drawn from the test set MIPLIB [2]. This was for at least two reasons: first, our code is not nearly as sophisticated as OSL in the way it chooses branching variables or the next subproblem to attack; secondly, these problems are just not large enough (none of them has more than 820 constraints) — problems that are amenable to solution with current hardware are not really big enough for the interior point method to perform better than the simplex method on the subproblems.

The paper [4] concerns the related problem of solving a mixed integer nonlinear

programming problem using branch-and-bound. Until now, all such branch-and-bound algorithms had solved the relaxations to optimality. We were able to demonstrate the advantage of branching early. We applied our algorithm to problems whose relaxations were convex problems and we used Lagrangian techniques to bound the values of the subproblems.

6 A primal-dual cutting plane method

We have developed a cutting plane algorithm based upon the primal-dual barrier method for linear programming [19, 20]. We have tested this algorithm on the linear ordering problem with good computational results.

One advantage of using the primal-dual barrier method is that both the primal and the dual solutions get updated at every iteration. Compared to the algorithms described in [22, 17], this provides considerably more flexibility over when to call the separation routines. We chose to call the cutting plane routines whenever the duality gap fell below a dynamically adjusted tolerance. If the separation routines are called too soon then superfluous constraints are added to the formulation; if they are called too late then additional iterations are spent on the current relaxation and the initial solution to the next relaxation is close to the boundary of the feasible region, which is not an advantageous starting point. The tolerance is adjusted dynamically depending on the number of violated cutting planes and their violations: if many constraints are found which are violated by large amounts then the tolerance is increased; if only a few constraints which are violated by a small amount are found then the tolerance is decreased.

We now describe the method we developed for restarting the algorithm after adding cutting planes. We implemented the algorithm so that the relaxation of the linear ordering problem was the primal problem. The cutting planes are inequality constraints in the primal problem, so their addition results in the addition of a slack variable to the primal problem; this slack variable must lie between zero and two for any point in S, the convex hull of the set of linear orderings. When cutting planes are added, columns are added to the dual problem but the corresponding variables have no sign constraint, so the new dual variables can be given the value zero and the new dual point is a feasible point for starting the interior point method. There is an additional dual constraint corresponding to the additional primal slack variable; this constraint contains both a slack and a surplus variable, so it can be satisfied by setting both of these additional dual variables equal to 0.1. The only other modification we made to the dual was to increase slack and surplus variables which are very close to their bounds — this was done for numerical stability. The addition of cutting planes makes the current primal solution infeasible. In our initial experiments

we attempted to use the infeasible interior point method described in, for example, Lustig et al. [16]. However, we found that the algorithm took several iterations to regain feasibility, by which time the solution was far from optimality in the current relaxation. In order to avoid this problem, we constructed a point x^{FEAS} which is in the relative interior of S. This point is always an interior point in any relaxation, so it is always feasible in the primal problem. Thus, when we restarted the interior point method after adding cutting planes, we set the primal point to x^{FEAS}. Initially x^{FEAS} is set to the vector of halves. We attempt to update it at every primal iteration: if the new primal solution is in S then x^{FEAS} is updated to this new solution; otherwise, x^{FEAS} is updated by taking a step from x^{FEAS} towards the new primal point. The work spent updating x^{FEAS} was minimal, but the number of iterations saved was considerable. A more sophisticated updating scheme for x^{FEAS} may well be justified. The only other update to the primal solution when adding cutting planes was to increase variables which were very close to their bounds, as was done for the dual solution; this was necessary for numerical stability.

We found it computationally efficient to drop constraints which were far from being tight. In this way, the linear algebra required to calculate the projections is kept tractable. The disadvantage of dropping constraints is that some constraints may be added and dropped repeatedly; in practice, this disadvantage is easily offset by the speed up in the linear algebra.

There was one other aspect of the implementation which was designed to make the linear algebra efficient. When we added cutting planes, we ensured that the set of additional constraints contributed at most one extra nonzero to any column of A. We found that this considerably improved the performance of the algorithm because the work required to calculate a projection was greatly decreased. Because of this choice, many more relaxations are examined and more iterations are taken, but the cost of this is easily outweighed by the decrease in the time required each iteration.

As in [22], an attempt was made to find feasible integer points by rounding the current point in the relaxation. The rounding heuristics depend on the particular combinatorial optimization problem. For the linear ordering problem, we attempted to find an ordering which corresponds closely to the fractional feasible point.

Grötschel et al. [12] implemented a simplex based algorithm for the linear ordering problem. It is impossible to really compare their runtimes with ours, because of the difference in hardware. Nonetheless, it does appear that our run times, at least for the larger, 56 and 60 sector, problems, are similar to theirs.

7 Conclusions

This research has shown that interior point methods can be adapted to solve integer programming problems. Because of the size of problems that can currently be solved, the methods we have developed have been at best competitive with methods based upon the simplex algorithm. Experience with linear programming problems has shown that the time required by an interior point method usually increases less quickly than that required by the simplex method as problem size increases. If a similar trend holds for integer programming problems (our results to date appear to show this same pattern), then we believe that these results hold great promise for the future, because better hardware will make larger problems tractable. We intend to continue developing these methods, trying them out on larger problems.

References

[1] J. E. Beasley. OR-library: Distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41(11):1069–1072, 1990.

[2] R. E. Bixby, E. A. Boyd, and R. R. Indovina. MIPLIB: A test set of mixed integer programming problems. *SIAM News*, 25(2):16, March 1992.

[3] B. Borchers. *Improved branch and bound algorithms for integer programming.* PhD thesis, Rensselaer Polytechnic Institute, Mathematical Sciences, Troy, NY, 1992.

[4] B. Borchers and J. E. Mitchell. An improved branch and bound algorithm for mixed integer nonlinear programming. Technical Report 200, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, August 1991.

[5] B. Borchers and J. E. Mitchell. Using an interior point method in a branch and bound algorithm for integer programming. Technical Report 195, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, 1991. Revised July 7, 1992.

[6] S. Chopra, E. R. Gorres, and M. R. Rao. Solving the Steiner tree problem on a graph using branch and cut. *ORSA Journal on Computing*, 4(3):320–335, 1992.

[7] A. S. El-Bakry, R. A. Tapia, and Y. Zhang. A study of indicators for identifying zero variables in interior-point methods. Technical Report TR-91-15, Dept. of Mathematical Sciences, Rice University, Houston, TX 77251, USA, 1991.

[8] R. M. Freund. A potential-function reduction algorithm for solving a linear program directly from an infeasible 'warm start'. *Mathematical Programming*, 52:441–466, 1991.

[9] C. C. Gonzaga. On lower bound updates in primal potential reduction methods for linear programming. *Mathematical Programming*, 52(3):415–428, 1991.

[10] M. Grötschel and O. Holland. Solving matching problems with linear programming. *Mathematical Programming*, 33:243–259, 1985.

[11] M. Grötschel and O. Holland. Solution of large-scale travelling salesman problems. *Mathematical Programming*, 51(2):141–202, 1991.

[12] M. Grötschel, M. Jünger, and G. Reinelt. A cutting plane algorithm for the linear ordering problem. *Operations Research*, 32:1195–1220, 1984.

[13] M. Grötschel, C. L. Monma, and M. Stoer. Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints. *Operations Research*, 40(2):309–330, 1992.

[14] D. den Hertog and C. Roos. A survey of search directions in interior point methods for linear programming. *Mathematical Programming*, 52(3):481–510, 1991.

[15] IBM. *IBM Optimization Subroutine Library Guide and Reference*, August 1990. Publication number SC23-0519-1.

[16] I. J. Lustig, R. E. Marsten, and D. F. Shanno. On implementating Mehrotra's predictor-corrector interior point method for linear programming. *SIAM Journal on Optimization*, 2:435–449, 1992.

[17] J. E. Mitchell. An interior point column generation method for linear programming using shifted barriers. Technical Report 191, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180–3590, 1990.

[18] J. E. Mitchell. Updating lower bounds when using Karmarkar's projective algorithm for linear programming. Technical Report 190, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180–3590, November 1990. Revised: June, 1991. To appear in *Journal of Optimization Theory and Applications 77(3), June 1993.*

[19] J. E. Mitchell and B. Borchers. A primal-dual interior point cutting plane method for the linear ordering problem. Technical Report 204, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180–3590, August 1992. To appear in *COAL Bulletin.*

[20] J. E. Mitchell and B. Borchers. Solving real-world linear ordering problems using a primal-dual interior point cutting plane method. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180–3590, 1992. (In preparation).

[21] J. E. Mitchell and M. J. Todd. A variant of Karmarkar's linear programming algorithm for problems with some unrestricted variables. *SIAM Journal on Matrix Analysis and Applications*, 10(1):30–38, 1989.

[22] J. E. Mitchell and M. J. Todd. Solving combinatorial optimization problems using Karmarkar's algorithm. *Mathematical Programming*, (to appear). January 27, 1989. Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180–3590, Revised May 17, 1990, and April 15, 1991.

[23] M. W. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33(1):60–100, 1991.

[24] M. J. Todd. Improved bounds and containing ellipsoids in Karmarkar's linear programming algorithm. *Mathematics of Operations Research*, 13(4):650–659, 1988.

[25] M. J. Todd and B. P. Burrell. An extension of Karmarkar's algorithm for linear programming using dual variables. *Algorithmica*, 1:409–424, 1986.